2.32 COMMAND GUIDANCE

SEE CLASSIFIED ADDENDUM

This document describes the RF proportional navigation missile systems that are in ESAMS. **SEE CLASSIFIED ADDENDUM**

This document thoroughly describes proportional navigation and **SEE CLASSIFIED ADDENDUM**

Fundamentals of Proportional Navigation

Proportional navigation works on the simple principle that the missile and target are on a collision course as long as the target line-of-sight rate has been nulled and the two are closing. As illustrated in Figure 2.32-1, this condition would mean that, for a semi-active missile system, the seeker is continually pointing in the same inertial direction while tracking the target.

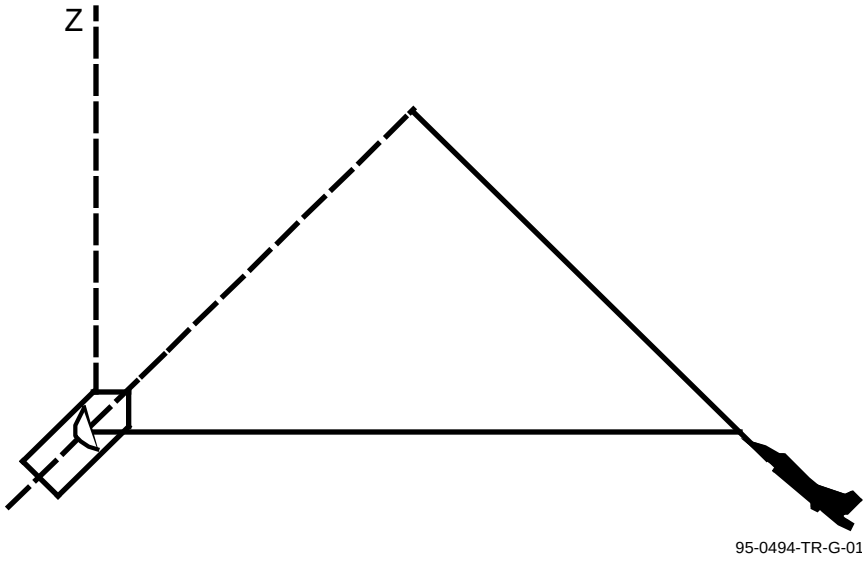


FIGURE 2.32-1. Target-Missile Intercept Geometry.

As with command guided simulations, proportional navigation systems may be modeled at different levels of fidelity. The core of this guidance concept is the seeker, and, in a simple form, it is represented by the circuit in Figure 2.32-2 and the transfer functions in equations (1) and (2).

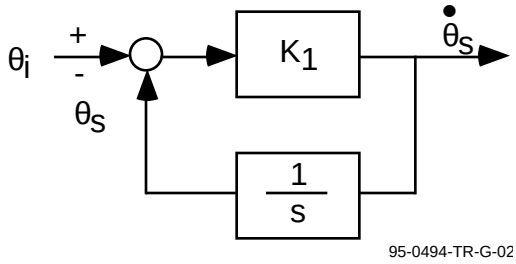


FIGURE 2.32-2. Simplified Seeker Circuitry.

$$\frac{\dot{\theta}_s}{\theta_I} = \frac{K_1}{1 + \frac{K_1}{s}} = \frac{K_1 s}{s + K_1} \qquad [2.32\text{-}1]$$

$$\frac{\dot{\theta}_s}{\dot{\theta}_I} = \frac{K_1}{s + K_1} = \frac{1}{1 + \frac{s}{K_1}} = \frac{1}{\tau s + 1} \qquad [2.32\text{-}2]$$

Thus, in the simple representation, where s is the Laplace operator d/dt, the relation between the developed seeker rate ($\dot{\theta}_{Is}$) and the input target rate ($\dot{\theta}_I$) is given by a first order time lag. The generated seeker rate will then represent the approximate target line-of-sight rate. Two constraints on $\dot{\theta}_s$ are the maximum allowable tracking rate and the boresight limits.

The integrated proportional navigation loop, with the simplified seeker representation, would appear as follows:

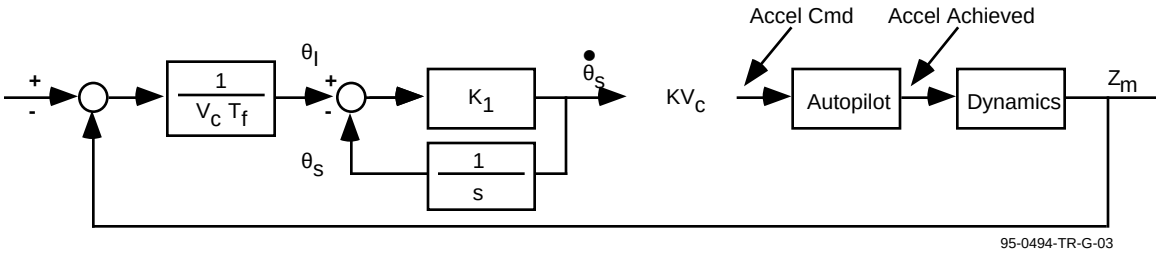


FIGURE 2.32-3. Simplified Pro Nav Guidance Loop.

Using small angle approximations in one plane, the target angle (θ_I) is represented as:

$$\theta_I \approx \frac{Z_t - Z_m}{V_c T_f} \qquad [2.32\text{-}3]$$

where

Z_t = Target Position (Z Coordinate)
Z_m = Missile Position (Z Coordinate)
V_cT_f = Closing Velocity * "Time-to-go" (Horizontal Separation)(4)

The guidance loop gain for a proportional navigation system is $KV_c\ \dot{\theta}_s$, where K is navigation gain, V_c is closing velocity, and $\dot{\theta}_s$ is target line-of-sight rate as measured by the seeker. Thus, if $\dot{\theta}_s$ goes to zero, the missile and target are on a collision course, and a zero acceleration command is provided to the autopilot. If $\dot{\theta}_s$ is not zero, the command is weighted by the navigation constant K and an estimate of closing velocity. Setting K equal to 3 has been shown to be optimal for some engagement scenarios (reference 2).

The closing velocity V_c (see (4) above) can be measured in a radar missile system by calculating the Doppler shift between the target and missile. By setting guidance loop gain based on this parameter, the missile performance will be more consistent over variations in closing velocity, since the inherent dependency of the target angle on closing velocity will be canceled in the overall loop gain. With infra-red missiles, closing velocity data are unavailable, and a nominal value must be used.

It would also be desirable to remove the guidance loop gain dependency on time-to-go (T_f). Some advanced guidance theories are based on an accurate estimate of T_f. However, to date, the difficulty appears to outweigh the payoff of implementing the advanced guidance.

Simplified models often make assumptions that are marginal. For example, some homing, or semi-active, missile simulations assume that the seeker is completely isolated from any missile pitch rate corruption. Thus, the seeker line-of-sight rate provided for use by the proportional navigation guidance law is perhaps more accurate than it should be in simplified simulations. It must be remembered that the target inertial line-of-sight rate is desired for use in the guidance command generation. Any corruption due to missile pitching will degrade missile intercept capability.

Figure 2.32-4 represents a semi-active proportional navigation missile with some generic circuitry that is used to isolate the seeker from missile pitching effects. The isolation circuitry works in the following manner. The target line-of-sight angle at the left is perturbed by the missile pitch angle, which is also fed into the stabilization circuitry. The stabilization loop, as shown in Figure 2.32-4, is a type 1 servo. If the input pitch rate is a step function, this stabilization loop will follow the step with zero error in the steady state. Hence, for a step function in missile pitch rate (upward), a counter command will be sent to the seeker which will just offset the pitch rate magnitude. Thus, the seeker will continue to be pointed at the target but will not be tracking at the target inertial line-of-sight rate.

The target inertial line-of-sight rate is extracted by taking the difference of the target angle and the seeker pointing angle and dividing by the time constant T_1. Ideally, this rate will be dependent on target motion and isolated completely from missile pitching. The rate is filtered and multiplied by KV_c (Figure 2.32-4) to provide the commanded acceleration which is sent to the missile autopilot.

FIGURE 2.32-4. Pro Nav With Seeker Isolation Circuitry.

Two other notes about Figure 2.32-4 are in order. First, if the missile pitch rate is not a step function, the body rate corruption will not be completely canceled. Thus, the system has to be designed to provide the desired accuracy under operational conditions.

Second, the circuitry as pictured in Figure 2.32-4 has the radome included. This means that the impact of a decrease in the stability margin due to the radome can be determined. Unfortunately, very little radome information is available to the community for use in digital simulations. Therefore, radome modeling is not usually included.

2.32.1 Functional Element Design Requirements

SEE CLASSIFIED ADDENDUM

1. Model the seeker stability and control circuitry.
2. Model the proportional navigation guidance command generation.

These requirements must be implemented in sufficient fidelity to reflect the true missile intercept capability over the entire **SEE CLASSIFIED ADDENDUM**

2.32.2 Functional Element Design Approach

This section describes the design approach (equations, algorithms, and methodology) implementing the design requirements of the previous section.

Figure 2.32.2-1 illustrates the **SEE CLASSIFIED ADDENDUM**

Design Element 32-1: Seeker Stability and Control Circuitry

The seeker stability and control circuitry is shown in Figure 2.3.2.2.1-1. As mentioned earlier, the system has to compensate for the missile pitching motion and the angular

movement of the target aircraft in order for the seeker to remain pointed at the target. The target inertial line-of-sight rate must then be extracted and used i the guidance command generation.

The target angles (σ_A azimuth, σ_E elevation) are introduced at the left of Figure 2.3.2.2.1-1. The missile pitching rates (BTMDTS elevation, PTMDTS azimuth) are also introduced at the left and drive rate gyros to provide output signals proportional to rates. Thus, at the junction points (YE06, YA06), signals are developed to drive the seeker system that are dependent both on target motion and missile ptiching. The result is that the seeker can be pointed continuously at the target.

Seeker stabilization is shown at the right of Figure 2.32.2.1-1. It has two feedback paths which are designed to provide stability and furnish the correct seeker system transient and steady state responses.

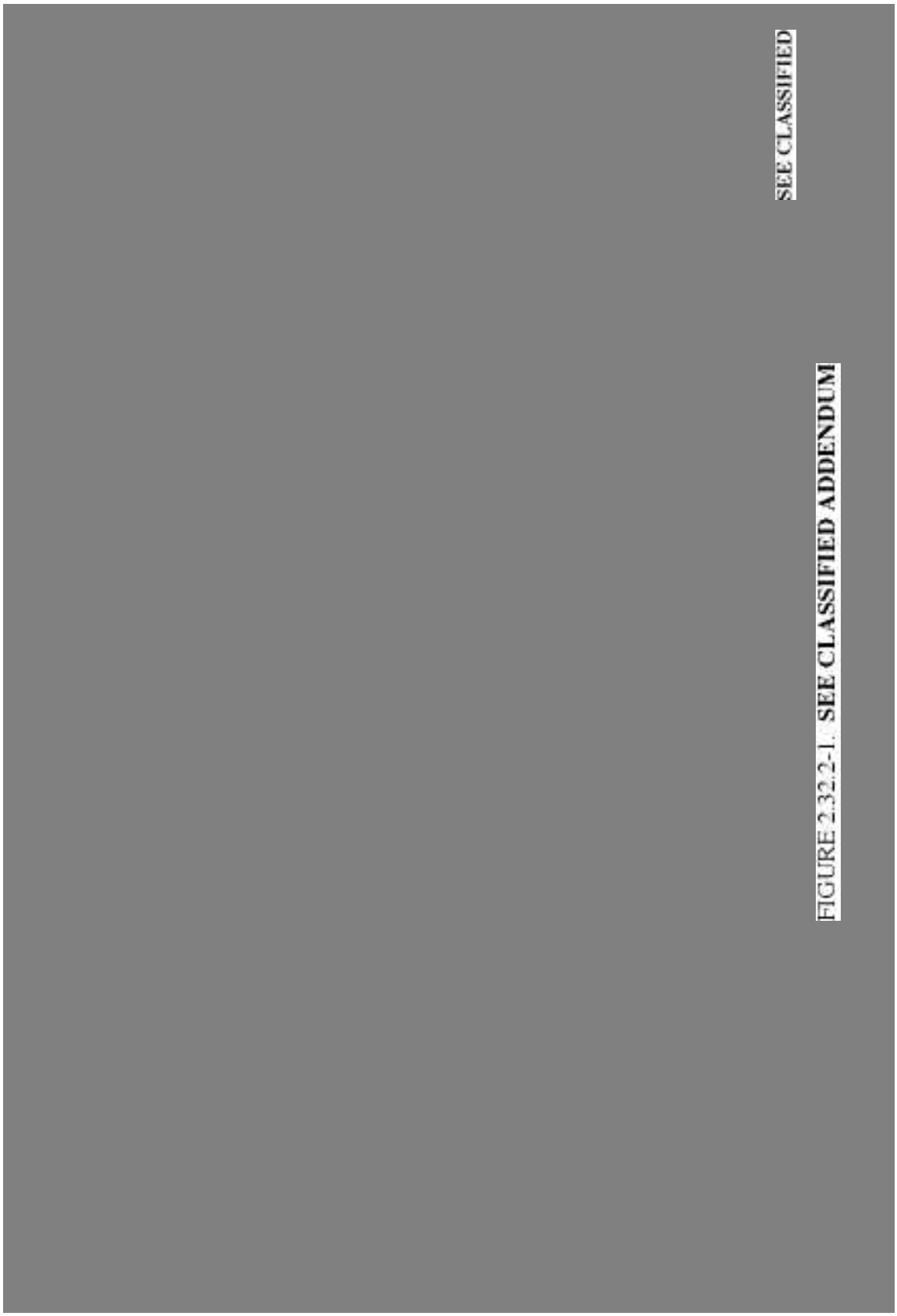


FIGURE 2.32.2-1. SEE CLASSIFIED ADDENDUM

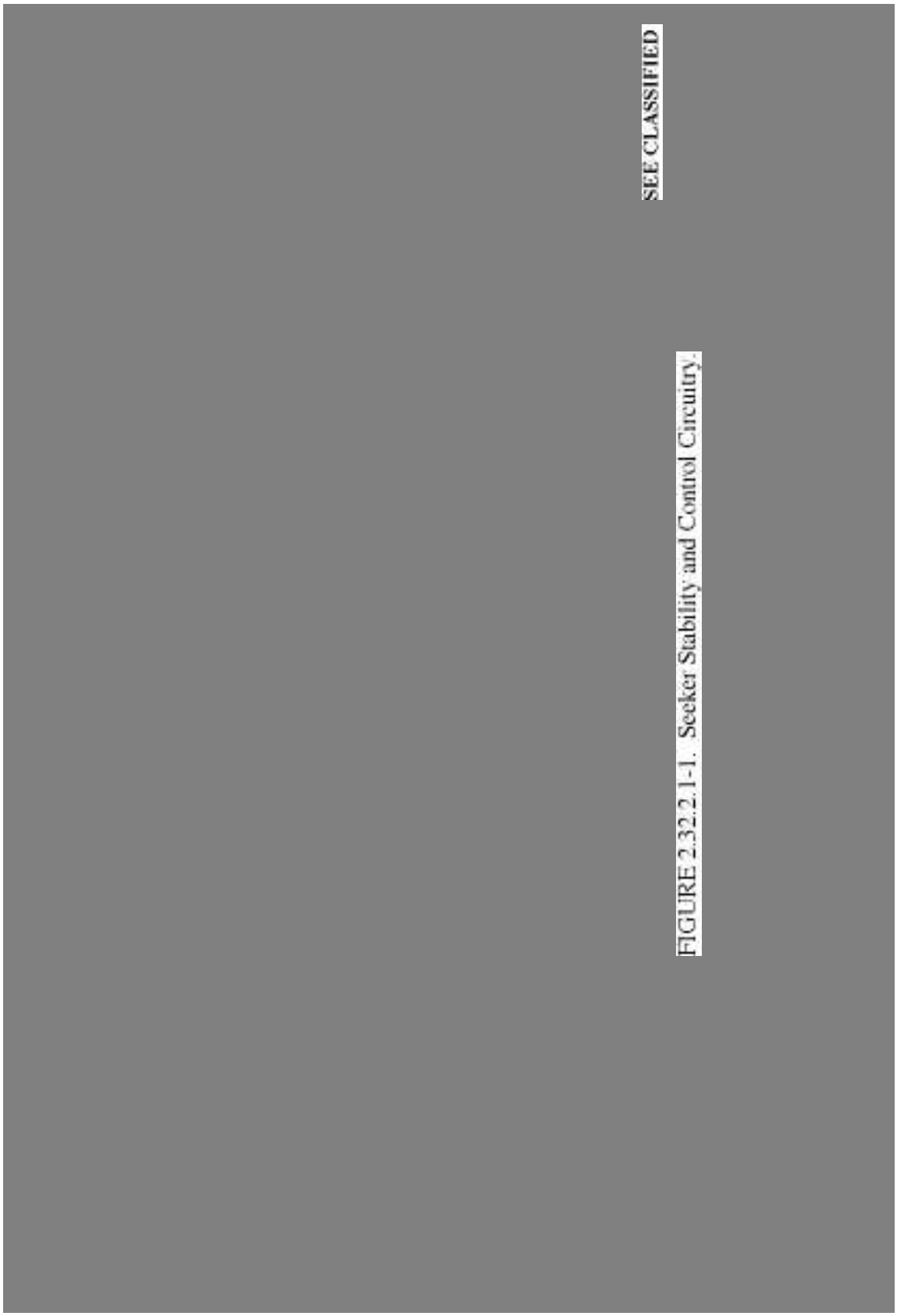


FIGURE 2.32.2.1-1. Seeker Stability and Control Circuitry.

SEE CLASSIFIED ADDENDUM

[2.32-4]

[2.32-5]

[2.32-6]

SEECLASSIFIEDADDENDUM [2.32-7]

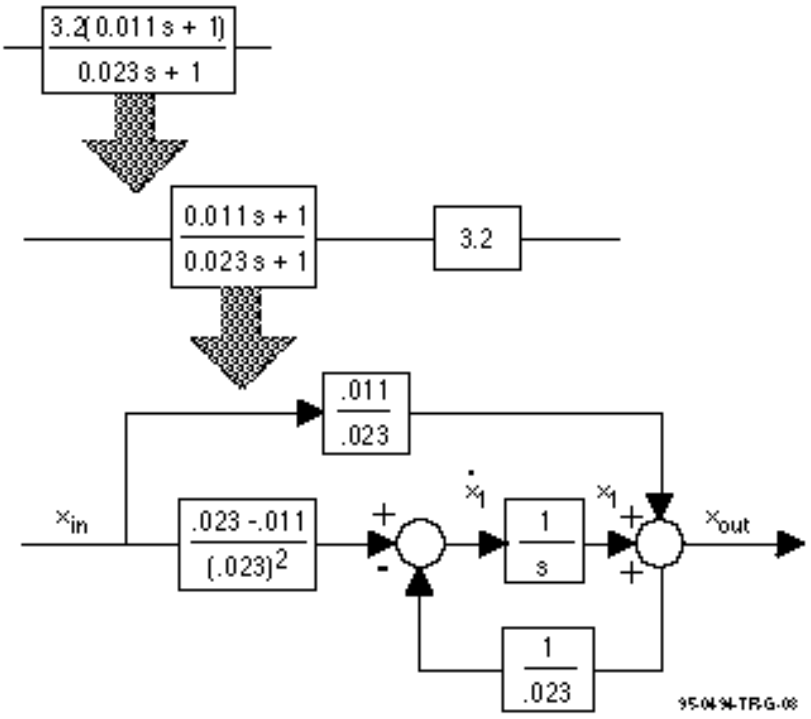
SEECLASSIFIEDADDENDUM [2.32-8]

SEECLASSIFIEDADDENDUM [2.32-9]

The output of the transfer function is

$$y = y_1 + K_O u = y_1 + .0264u \quad [2.32\text{-}10]$$

Filter 3 is developed as follows once again using state space representation:



$$x_{out} = x_1 + (.011/.023) * x_{in} \quad [2.32\text{-}11]$$

Block 4 is similar in structure to block 3.

$$x_{out} = x_1 + 61.3\ (0.023/0.1) * x_{in} \quad [2.32\text{-}12]$$

The form developed above comes about as follows:

$$\frac{y}{u} = \frac{\tau_1 s + 1}{\tau_2 s + 1} = \frac{\frac{\tau_1}{t_2} s + \frac{1}{\tau_2}}{s + \frac{1}{\tau_2}} \qquad [2.32\text{-}13]$$

$$\frac{y}{u} = \frac{\tau_1}{\tau_2} + \frac{\frac{1}{\tau_2} - \frac{\tau_1}{\tau_2^2}}{s + \frac{1}{t_2}} = \frac{\tau_1}{\tau_2} + \frac{\frac{\tau_2 - \tau_1}{\tau_2}}{s + \frac{1}{\tau_2}} \qquad [2.32\text{-}14]$$

hence

$$y = \frac{\tau_1}{\tau_2} u + \frac{\left(\frac{\tau_2 - \tau_1}{\tau_2}\right)}{\tau_2 s + 1} u \qquad [2.32\text{-}15]$$

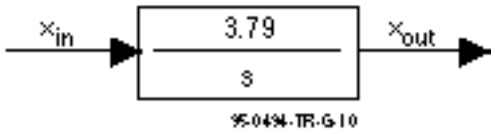
Let

$$x = \left(\frac{\tau_2 - \tau_1}{\tau_2^2}\right) / \tau_2 s + 1 \div u \qquad [2.32\text{-}16]$$

and

$$\dot{x} = \frac{1}{\tau_2}\left(\left(\frac{\tau_2 - \tau_1}{t_2}\right) u - x\right) \qquad [2.32\text{-}17]$$

Block 5 is a gain and an integrator and has the form



[2.32-18]

$$\dot{x}_{out} = 3.79 * x_{in} \qquad [2.32\text{-}19]$$

$$x_{out} = x_{out} + \dot{x}_{out}\ dt \qquad [2.32\text{-}20]$$

where dt is a time increment.

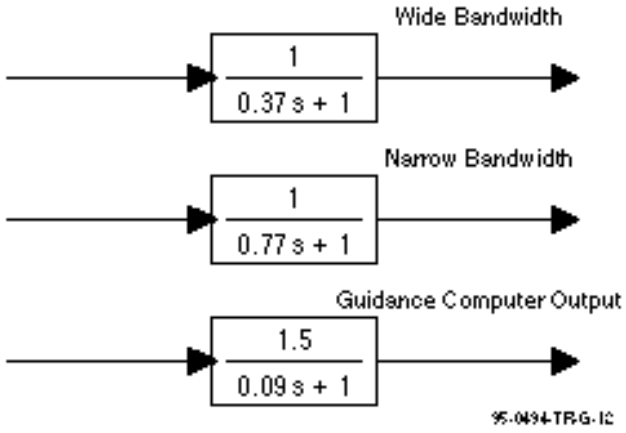
The above equations have presented the general procedure in executing the transfer function dynamics, as stated previously, fourth order Runge-Kuttas provide the actual integrations.

Design Element 32-2: Guidance Command Generation

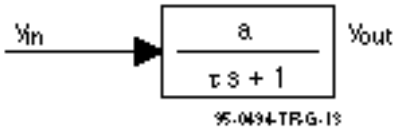
Figure 2.32.2.2-1 shows the guidance command development. At the left of the figure, seeker rates and body rates are introduced, and they are multiplied by a gain factor (K_T and K_G) that is dependent on closing velocity. As mentioned previously, this procedure allows dependency of the guidance loop gain on closing velocity to be deleted. Based on sign convention, the body rates are subtracted from seeker rates so that inertial target information is used. Gains are then used as appropriate to develop the standard proportional navigation law pnc $*V_c * \dot{\sigma}$.

The guidance circuity also has a **SEE CLASSIFIED ADDENDUM**

The three filtering transfer functions are



These three transfer functions can be represented by



[2.32-21]

$$\tau \dot{y}out + yout = a\ yin \qquad [2.32\text{-}22]$$

$$\dot{y}out = \frac{1}{\tau}(a\ yin - yout) \qquad [2.32\text{-}23]$$

$$\dot{y}out = \tau_1(a\ yin - yout) \qquad [2.32\text{-}24]$$

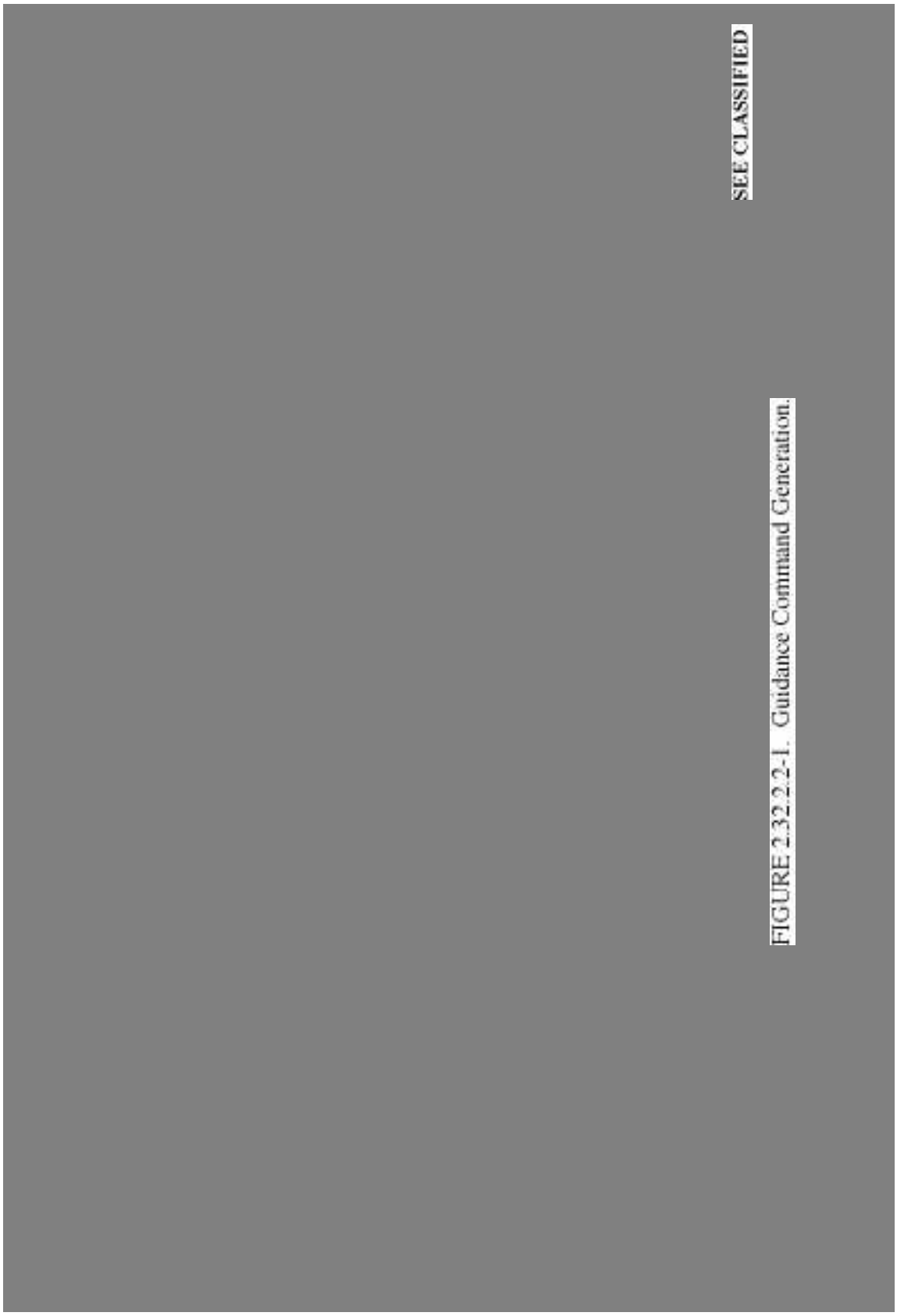


FIGURE 2.32.2.2-1. Guidance Command Generation.

2.32.3 Functional Element Software Design

This section describes the software design necessary to implement the functional element requirements for proportional navigation, as outlined in section 2.32.1, and the design approach, as outlined in section 2.32.2. Section 2.32.3 is organized in four subparts: The first subpart gives the overall subroutine hierarchy and gives capsule descriptions of the relevant subroutines; the second subpart contains a functional flow chart for the functional element as a whole, and describes the major operations represented by each block in the chart; the third subpart presents detailed logical flow charts for the subroutines; and the last subpart contains a description of all input and output for the functional element as a whole and for each subroutine that implements the functional element.

Proportional Navigation Subroutine Hierarchy

The Fortran call tree that is implemented for the **SEE CLASSIFIED ADDENDUM**

The subroutine **SEE CLASSIFIED ADDENDUM**

TABLE 2.32-1. SEE CLASSIFIED ADDENDUM

Module Name	Description
CONSYS	Selects system-specific guidance control routine ("GAP" routine" according to the value of the guidance type index IGTYPE.
GAPJ	A specific guidance control routine that sets up guidance integration control variables, calls the integrator, and converts the resulting fin deflections to voltages for use by the autopilot for this system.
RK4G	General fourth-order Runge Kutta integration routine that integrates the guidance computer state equations one time step, using the guidance integration control variables and the guidance derivative routine ("DRVG" routine) specified by the GAP routine.
RELTGT	Computes the target-missile relative geometry at the integration time-points for RK4G.
DRVGJ	System-specific guidance derivative routine that implements computation of the system's seeker and guidance computer state variables.

Proportional Navigation Functional Flow

Figure 2.32.3-2 shows the top-level functional flow of the Proportional Navigation Guidance implementation. The actual guidance circuitry is fully contained in subroutine DRVGJ. Hence, its flow is captured in Figure 2.32.3-2.

Figure 2.32.3-2 illustrates the flow of. . .**SEE CLASSIFIED ADDENDUM**

In this figure, the blocks are numbered and they are discussed below:

Block 1. The seeker angles are initialized in the right direction by obtaining the appropriate target parameters.

Block 2. **SEE CLASSIFIED ADDENDUM**

Block 3. **SEE CLASSIFIED ADDENDUM**

Block 4. Subroutine ANGTRN is called to calculate the seeker pointing errors in the frame rolled 45° to the normal azimuth and elevation directories.

Block 5. Either a wide or narrow bandwidth can be selected based on dynamic pressure considerations.

Block 6. Gain in the target input loop has a non-linear gain feature.

Block 7. Both target motion and missile pitching are input to the seeker drive circuitry so that the seeker continues to track the target in the presence of missile notion.

Block 8. The signal path includes two filters for signal smoothing before the processing is accomplished by the seeker circuitry.

Block 9. The input to the seeker servo amplifiers is composed of the signal path described above and feedback loops. The feedback is used to develop the correct transient and steady-state responses.

Block 10. Seeker rates are developed based on closed loop response. The rates are crucial to guidance command development.

Block 11. Seeker rates are integrated to obtain seeker pointing angles. These angles are used to obtain off-boresight target data so that the signal processing algorithms can extract the new target angle estimates.

Block 12. Seeker rates and missile body rates are fed into the guidance loops so that target inertial rates are extracted to drive the guidance command generation. Gains K_T and K_G are used to provide closing velocity data for use in the guidance command generation.

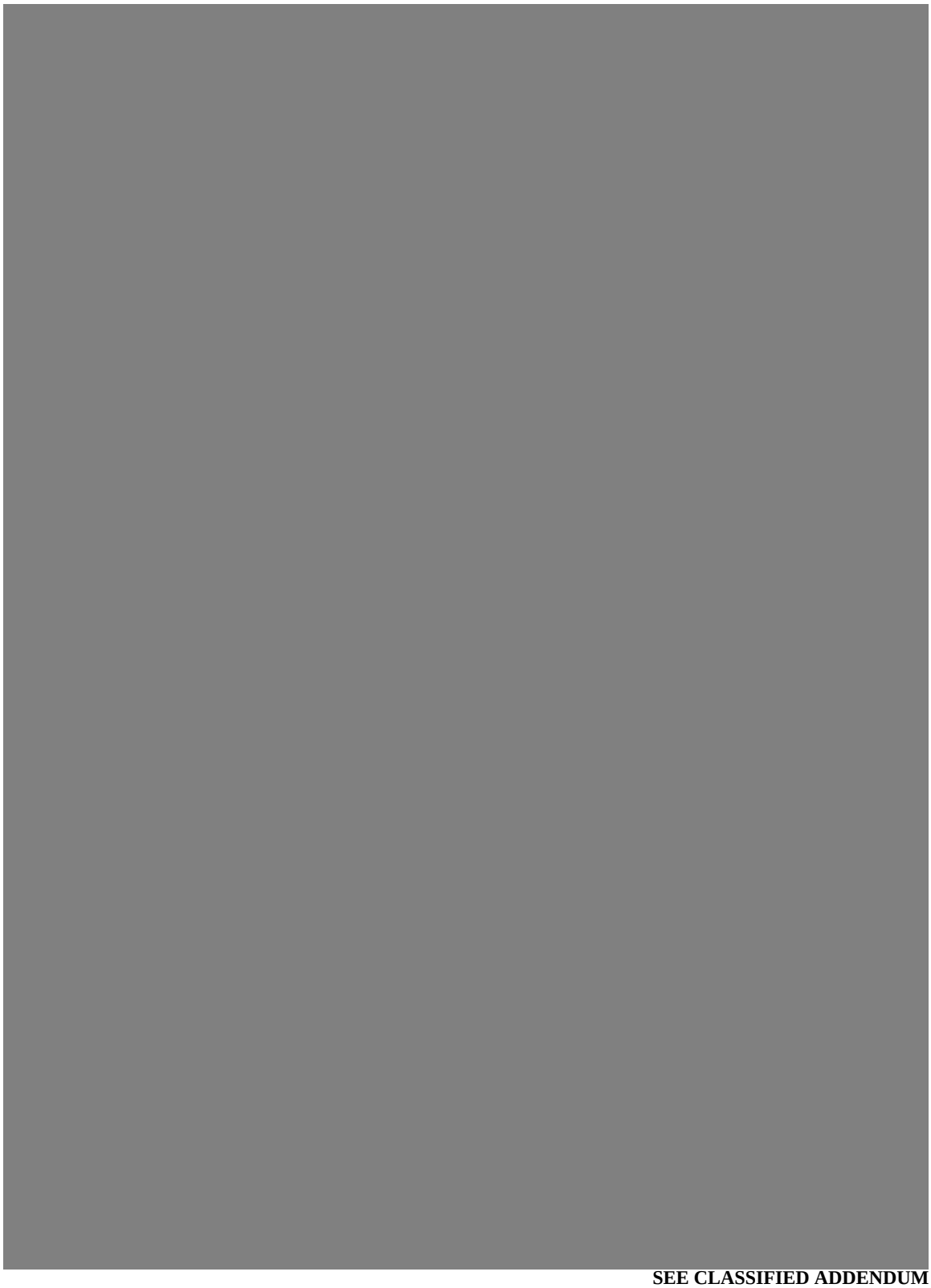
Block 13. Gain K_1 is determined and used in conjunction with wide and narrow band guidance computations.

Block 14. Guidance loop gains are established based on dynamic pressure and time-to-arm considerations.

Block 15. Guidance computer outputs determine guidance commands in volts.

Block 16. Guidance commands are converted to g's for use by the autopilot.

Block 17. Derivatives of the state variables are updated for use by RK4G in getting new values of the state variables.



SEE CLASSIFIED ADDENDUM

FIGURE 2.32-5. SEE CLASSIFIED ADDENDUM

Subroutine Flow Charts

The guidance circuitry for the **SEE CLASSIFIED ADDENDUM**

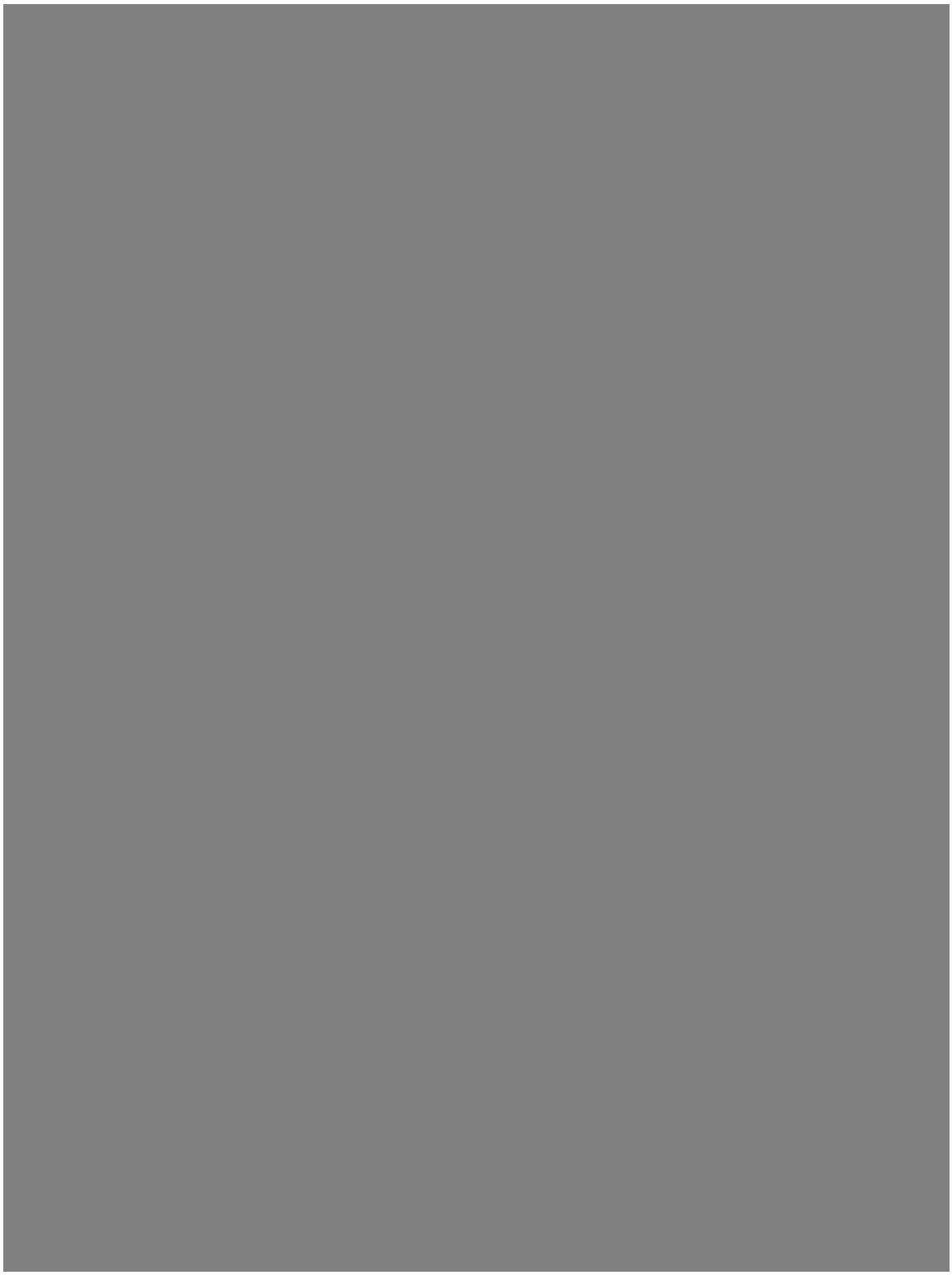


FIGURE 2.32-6. Functional Flow Diagram for Subroutine DRVGJ.

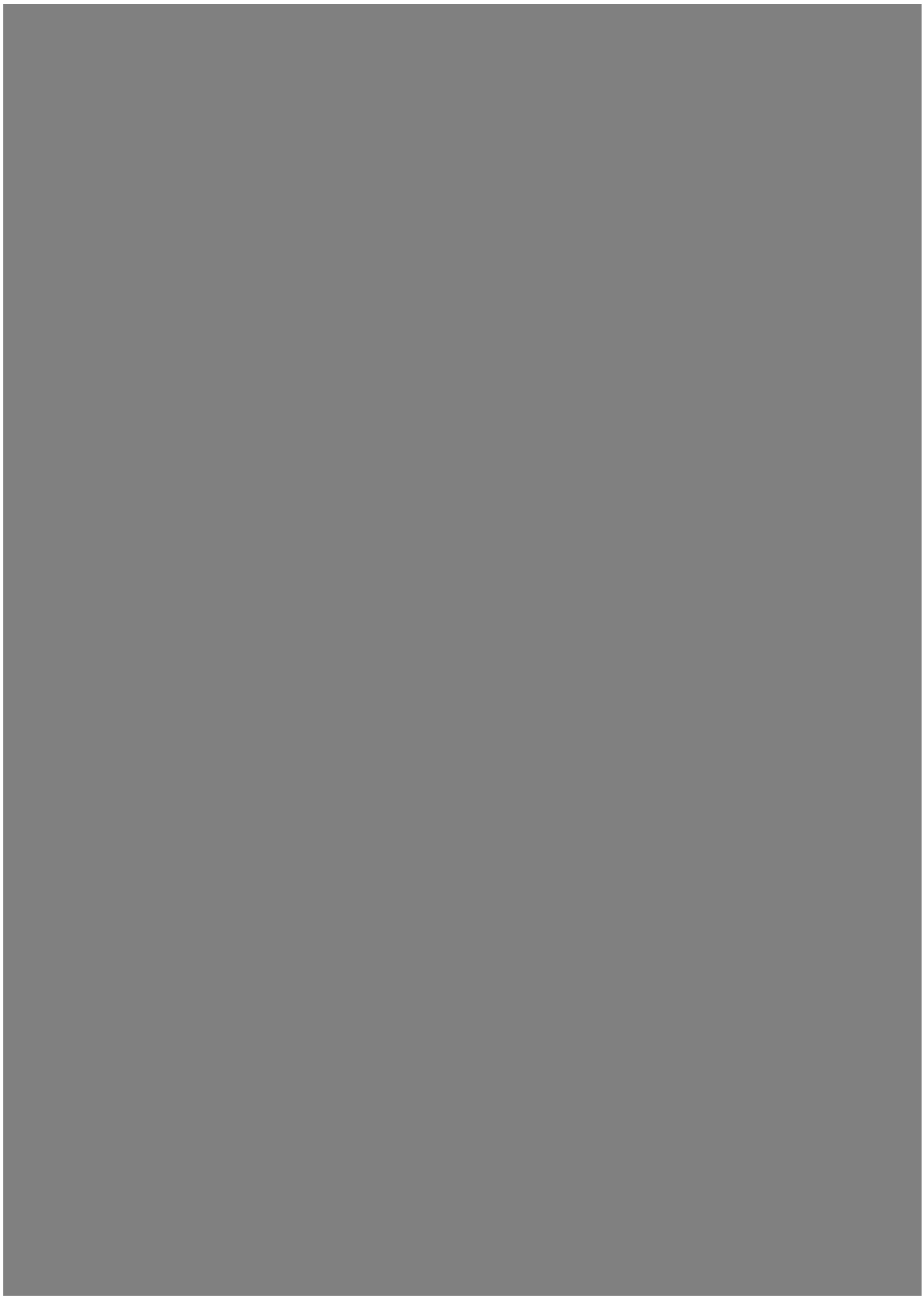


FIGURE 2.32-6. Functional Flow Diagram for Subroutine DRVGJ. (Contd.)

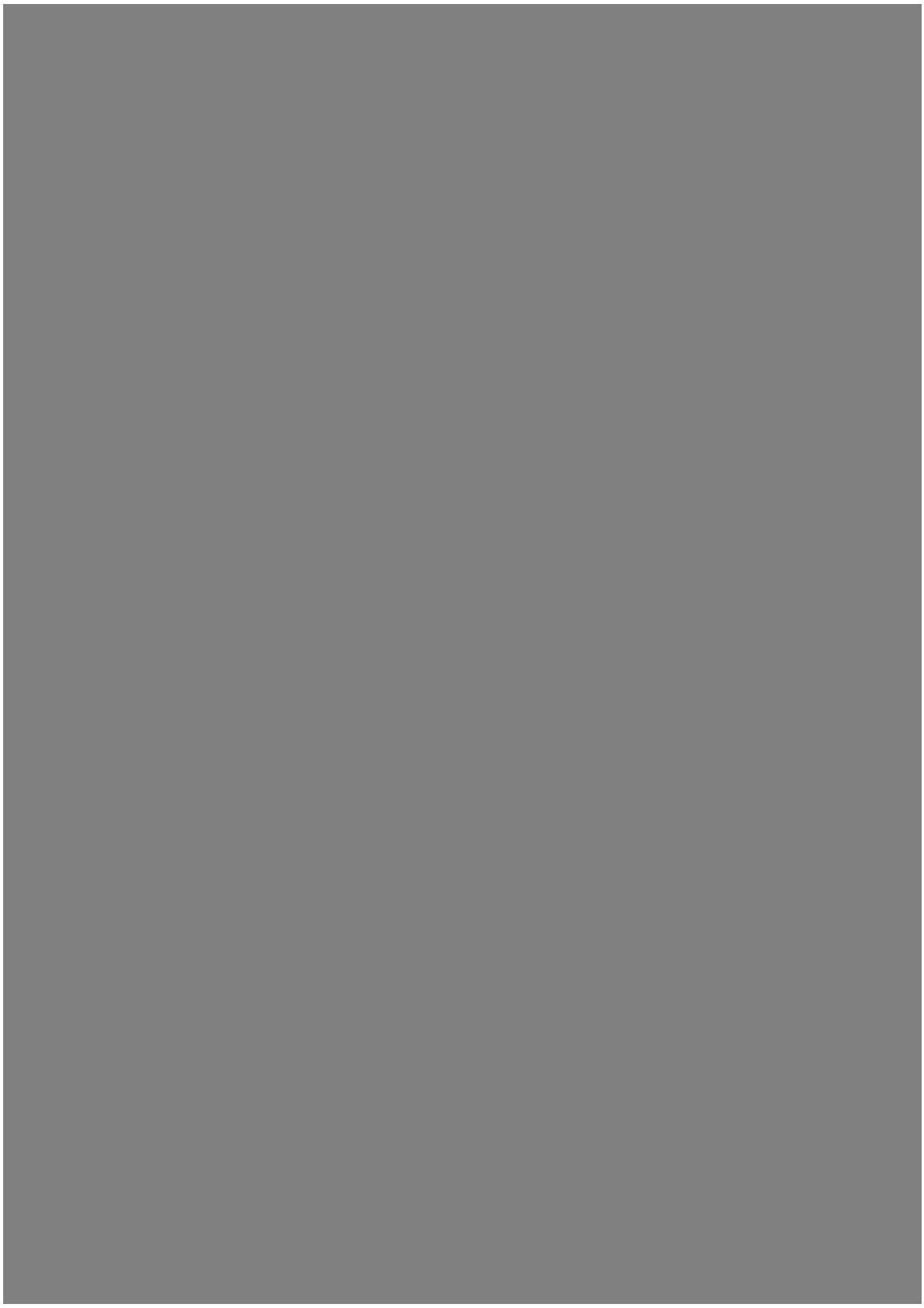


FIGURE 2.32-6. Functional Flow Diagram for Subroutine DRVGJ. (Contd.)

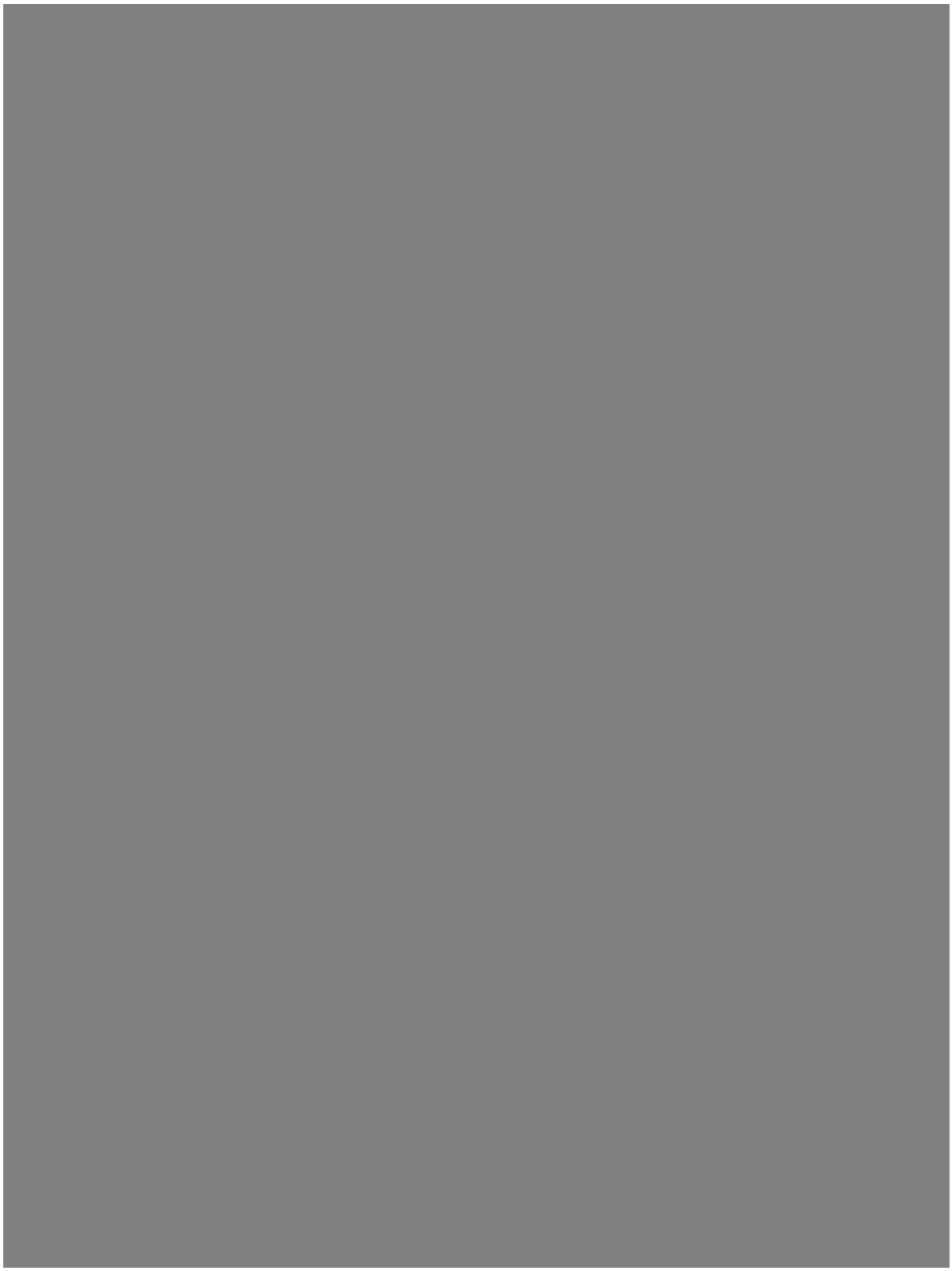


FIGURE 2.32-6. Functional Flow Diagram for Subroutine DRVGJ. (Contd.)

Proportional Navigation Guidance Inputs and Outputs

The model user inputs that affect the . . . **SEE CLASSIFIED ADDENDUM**

TABLE 2.32-2. User Inputs for the Proportional Navigation Guidance Functional Element.

Name	Kind	Description
DT	Common SIMVI	Length of time step for integration. Initialized from MSLD input DTM (missile time step) in subroutine SAMS.
GC	Common MSLD	Array of guidance coefficients; meaning of individual elements depends on specific system. See ESAMS Threat Manual for meanings specific to this system.
IGTYPE	Common ROPTN	Index indicating which guidance type to use.Initialized from MSLD input FLGMSL(3) in subroutine SAMI.
TBALL	Common MSLD	Time in missile flight that the missile will go ballistic (sec).

The outputs of the . . . **SEE CLASSIFIED ADDENDUM**

TABLE 2.32-3. Outputs of the Proportional Navigation Guidance Functional Element.

Name	Kind	Description
APITCH	Common GUIDAP	Current pitch acceleration command. Not used in all command guidance systems; in particular, not used by systems implemented by GAPJ/DRVGJ.
AYAW	Common GUIDAP	Current yaw acceleration command. Not used in all command guidance systems; in particular, not used by systems implemented by GAPJ/DRVGJ.
BOREAZ (ISEKR)	Common FREND	Seeker gimbal angles converted to inertial coordinate azimuth.
BOREEL (ISEKR)	Common FREND	Seeker gimbal angles converted to inertial coordinate elevation.
BTMDTS	Common GUIDAP	Seeker elevation rate (rad/sec).
DLT	Common SIMVI	Length of time step for integration.
EK1	Common GUIDAP	Autopilot command in fin plane 1.
EK2	Common GUIDAP	Autopilot command in fin plane 2.
GDCMP	Common GUIDAP	Limited elevation guidance command. Not used as output in all command guidance systems; in particular, not used as output by systems implemented by GAPJ/DRVGJ.
GDCMY	Common GUIDAP	Limited azimuth guidance command. Not used as output in all command guidance systems; in particular, not used as output by systems implemented by GAPJ/DRVGJ.
ISMODE	Common SIMVI	Flag to request seeker turn on, set by guidance and used by radar logic; not used by systems implemented by GAPJ/DRVGJ.
KST	Common SIMVI	Counter of number of full time steps; incremented by DRVGJ when INTFLG is one.
PGIMAN	Common GUIDAP	Pitch seeker gimbal angle (rad).
PTMDTS	Common GUIDAP	Seeker azimuth rate (rad/sec).
VGYROA	Common GUIDAP	Seeker elevation rate (BTMDTS) converted to a voltage signal.

TABLE 2.32-3. Outputs of the Proportional Navigation Guidance Functional Element.

Name	Kind	Description
VGYROE	Common GUIDAP	Seeker azimuth rate (PTMDTS) converted to a voltage signal.
XD	Argument	Array of derivatives to use in integration at end of current integration step; updated by the "DRVG" routine.
XI	Argument	Array of integrated variables at end of current integration step; updated by the "DRVG" routine.
YGIMAN	Common GUIDAP	Yaw seeker gimbal angle (rad).

SEE CLASSIFIED ADDENDUM

TABLE 2.32-4.

Input for Subroutine CONSYS		
Name	**Kind**	**Description**
FTIME	Common MISSIL	Current time from beginning of missile flight (sec).
IFATAL	PARAMETER, Include CONST	Symbolic name for value to indicate in error message that the error is a fatal kind.
IGTYPE	Common ROPTN	Index indicating which guidance type to use.
TBALL	Common MSLD	Time in missile flight that the missile will go ballistic (sec).

Output for Subroutine CONSYS		
Name	**Kind**	**Description**
APITCH	Common GUIDAP	Current pitch acceleration command. Not used in all command guidance systems; in particular, not used by systems implemented by GAPJ/DRVGJ.
AYAW	Common GUIDAP	Current yaw acceleration command. Not used in all command guidance systems; in particular, not used by systems implemented by GAPJ/DRVGJ.
EK1	Common GUIDAP (implicit)	Autopilot command in fin plane 1.
EK2	Common GUIDAP (implicit)	Autopilot command in fin plane 2.
ISMODE	Common SIMVI	Flag to request seeker turn on, set by guidance and used by radar logic; not used by systems implemented by GAPJ/DRVGJ.

Input for Subroutine GAPJ		
Name	**Kind**	**Description**
DT	Common SIMVI	Length of time step for integration.
FTIME	Common MISSIL	Current time from beginning of missile flight (sec).
GC	Common MSLD	Array of guidance coefficients; meaning of individual elements depends on specific system. Element GC(48) is the limit on magnitude of gimbal angles. See ESAMS Threat Manual for other meanings specific to this system.
GDCMP	Common GUIDAP	Pitch guidance command; set by call to DRVGJ via RK4G.
GDCMY	Common GUIDAP	Yaw guidance command; set by call to DRVGJ via RK4G.
ISEKR	PARAMETER, Include ARYBND	Symbolic flag identifying the seeker radar type.
PHI	Common EULER	Roll angle of missile.
PIO2	PARAMETER, Include CONST	Symbolic constant for pi over 2.
XD	Common INTEG	Array of derivatives to use in integration at start of current time step; also updated by the calls of the “DRVG” routine.
XI	Common INTEG	Array of integrated variables at start of current time step; also updated by the calls of the “DRVG” routine.

Output for Subroutine GAPJ		
Name	**Kind**	**Description**
BOREAZ (ISEKR)	Common FREND	Seeker gimbal angles converted to inertial coordinate azimuth; set by return argument by SKRANG.
BOREEL (ISEKR)	Common FREND	Seeker gimbal angles converted to inertial coordinate elevation; set by return argument by SKRANG.
DLT	Common SIMVI	Length of time step for integration.
DRVGJ	Argument to Subroutine RK4G	EXTERNAL name of subroutine to be used as derivative support in the integration by RK4G.
EK1	Common GUIDAP	Autopilot command in fin plane 1.
EK2	Common GUIDAP	Autopilot command in fin plane 2.
N	Argument to Subroutine RK4G	Number of variables for RK4G to integrate in this case.
PGIMAN	Common MISSIL	Pitch seeker gimbal angle (rad); set by return argument by ANGTRN.
T	Argument to Subroutine RK4G	Time since start of missile flight at start of current integration time step (sec); is advanced by one time step by RK4G.
Y	Common MISSIL	Y coordinate of the unit vector directed along the boreline of the missile seeker. Actually used only locally, but the variable is in the indicated common block.
YGIMAN	Common MISSIL	Yaw seeker gimbal angle (rad); set by return argument by ANGTRN.
Z	Common MISSIL	Z coordinate of the unit vector directed along the boreline of the missile seeker. Actually used only locally, but the variable is in the indicated common block.

Input for Subroutine RK4G		
Name	**Kind**	**Description**
DLT	Argument	Time step for Runge-Kutta integration.
DRV	Argument	Name of "DRVG" routine to call for derivative evaluation; declared EXTERNAL.
MAXXI	PARAMETER, Include ARYBND	Dimension of intermediate integral array XIP.
N	Argument	Number of variables to integrate.
SMALL	PARAMETER, Local.	Small number used to avoid underflows.
T	Argument	Time since start of missile flight at start of current integration time step (sec).
TOL	Common SUMARY	Time of missile launch (sec).
XD	Argument	Array of derivatives to use in integration at start of current time step; also updated by the calls of the "DRVG" routine.
XI	Argument	Array of integrated variables at start of current time step; also updated by the calls of the "DRVG" routine.

Output for Subroutine RK4G		
Name	**Kind**	**Description**
T	Argument	Time since start of missile flight at end of current integration time step (sec).
XD	Argument	Array of derivatives to use in integration at end of current time step.
XI	Argument	Array of integrated variables at end of current time step.

Input for Subroutine DRVGJ		
Name	**Kind**	**Description**
AZERR	Common ENVRN	Azimuth error due to multipath and clutter.
BOREEL (ITRKR)	Common FREND	Target tracking radar boresight pointing angle in elevation.
ELERR	Common ENVRN	Elevation error due to multipath and clutter.
EPSIG	Return Argument from ANGTRN	Phi-rolled yaw seeker tracking error (rad).
ETHETG	Return Argument from ANGTRN	Phi-rolled pitch seeker tracking error (rad).
FMACH	Common MISSIL	Current Mach number for missile.
GC	Common MSLD	Array of guidance coefficients; meaning of individual elements depends on specific system. See ESAMS Threat Manual for meanings specific to this system.
INTFLG	Argument	Runge Kutta integration step counter (1 to 4, corresponding to the four steps of fourth-order integration).
ITRKR	PARAMETER, Include ARYBND	Identifier of the tracker radar type.

Input for Subroutine DRVGJ		
Name	**Kind**	**Description**
KST	Common SIMVI	Counter of number of full time steps; incremented by DRVGJ when INTFLG is one.
NOGUID	Common SIMFLG	No-guidance flag, set by SKRGRP.
OMEG(-)	Common MISSIL	Missile body rates (rad/sec); 3 components.
PGIM	Return argument from ANGTRN	Phi-rolled pitch seeker gimbal angle (rad).
PGIMAN	Common GUIDAP	Pitch seeker gimbal angle (rad).
PHI	Common EULER	Missile roll angle (deg).
Q	Common MISSIL	Dynamic pressure (newton/m**2).
R2D	PARAMETER, Include CONST	Mathematical constant, conversion from radians to degrees.
RTMDOT	Return from Subroutine RELV	Range rate between missile and target at current time.
T	Argument	Time since start of missile flight at start of current integration time step (sec); is advanced by one-half and one time step by RK4G.
TIME	Common TARG	Current time (not flight time).
TOFF	Common GUIDAP	Time remaining to intercept (sec).
WYS	Return argument from B2SKR	Body rate in seeker frame (rad/sec), y-component.
WZS	Return argument from B2SKR	Body rate in seeker frame (rad/sec), z-component.
XD	Argument	Array of derivatives to use in integration at start of current integration step; is updated by the “DRVG” routine.
XI	Argument	Array of integrated variables at start of current integration step; is updated by the “DRVG” routine.
XMS	Common RELSIT	Missile-to-site separation, x-component
XT	Common TARG	Target location in inertial coordinate system, x-component
XTD	Return argument from TRKVEL	Target velocity in inertial coordinate system, x-component
YGIM	Return argument from ANGTRN	Phi-rolled yaw seeker gimbal angle (rad).
YGIMAN	Common GUIDAP	Yaw seeker gimbal angle (rad).
YMS	Common RELSIT	Missile-to-site separation, y-component
YT	Common TARG	Target location in inertial coordinate system, y-component
YTD	Return argument from TRKVEL	Target velocity in inertial coordinate system, y-component
ZMS	Common RELSIT	Missile-to-site separation, z-component
ZT	Common TARG	Target location in inertial coordinate system, z-component
ZTA	Return argument from TRKZ	Measured target altitude (m).
ZTD	Return argument from TRKVEL	Target velocity in inertial coordinate system, z-component

Output for Subroutine DRVGJ		
Name	**Kind**	**Description**
BTMDTS	Common GUIDAP	Seeker elevation rate (rad/sec).
GDCMP	Common GUIDAP	Limited elevation guidance command.
GDCMY	Common GUIDAP	Limited azimuth guidance command.
KST	Common SIMVI	Counter of number of full time steps; incremented by DRVGJ when INTFLG is one.
PGIMAN	Common GUIDAP	Pitch seeker gimbal angle (rad); updated as return argument from ANGTRN.
PTMDTS	Common GUIDAP	Seeker azimuth rate (rad/sec).
VGYROA	Common GUIDAP	Seeker elevation rate (BTMDTS) converted to a voltage signal.
VGYROE	Common GUIDAP	Seeker azimuth rate (PTMDTS) converted to a voltage signal.
XD	Argument	Array of derivatives to use in integration at end of current integration step; updated by the “DRVG” routine.
XI	Argument	Array of integrated variables at end of current integration step; updated by the “DRVG” routine.
YGIMAN	Common GUIDAP	Yaw seeker gimbal angle (rad); updated as return argument from ANGTRN.

2.32.4 Assumptions and Limitations

The seeker and guidance computer for the **SEE CLASSIFIED ADDENDUM**

No limitations are known at this time.

2.32.5 Functional Process Description

Overview of Module/Subroutine Functionality

Figure 2.32.5.1-1 displays an overview of the functionality that determines the results and impact of the **SEE CLASSIFIED ADDENDUM**

SEE CLASSIFIED ADDENDUM

FIGURE 2.32-7. **SEE CLASSIFIED ADDENDUM**

Discussion of the Implementation of Design Elements

Since the Proportional Navigation functional element implementation details are localized in the DRVG routine, which is discussed in detail in section 2.32.6.2, no additional discussion is given in this section.

Relationship of F.E. to the Whole Model

The **SEE CLASSIFIED ADDENDUM**

2.32.6 Annotated Code

Module Description

There are three general subroutines used to generate the

SEE CLASSIFIED ADDENDUM

Subroutine RK4G

This routine calls an external derivative routine for four derivative evaluations. It calls RELTGT before the calls to DRV at the beginning, mid point, and end point of the current time step to update the missile-target relationship. It combines the four derivatives by the Runge Kutta scheme to perform the integration.

Subroutine RELTGT

This routine will call TARGET to update the target position, velocity, and attitude. Then it will compute a new position and velocity of the missile and new target-missile-radar site relationship.

Subroutine DRVGJ

DRVGJ calculates the pitch and yaw acceleration commands currently valid from the integral array. The seeker tracking rates are calculated using simplified seeker dynamics. The returned rates are then used as inputs to the guidance computer, the dynamics of which are include dint his routine. The closing velocity is computed and used to determine the guidance gain. The derivatives of the state space equations which represent the seeker and guidance computer Laplace block diagram are integrated by the calling subroutine to solve for the parameters appearing in the block diagrams.

2.32.6.1 Annotated Source Code Listings

The

SEE CLASSIFIED ADDENDUM

Subroutine RK4G (contents are Unclassified)

The guidance routines in ESAMS use a standard fourth order Runge-Kutta integration algorithm. As identified in reference 4, the form is as follows.

$$Y_{n+1} = Y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}, \text{ for } \qquad \Delta x^5 \qquad [2.32\text{-}25]$$

$$k_1 = \Delta x f(x_n, y_n) \qquad [2.32\text{-}26]$$

$$k_2 = \Delta x f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{k_1}{2}\right) \qquad [2.32\text{-}27]$$

$$k_3 = \Delta x f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{k_2}{2}\right) \qquad [2.32\text{-}28]$$

$$k_4 = \Delta x f(x_n + \Delta x, y_n + k_3) \qquad [2.32\text{-}29]$$

In obtaining an updated value of a variable, four estimates of the increment in the variable during the program time step are obtained. The estimates are weighted as shown above, with one estimate made at the beginning point of the time-step, two at the mid-point of the time step, and one at the end point. A final consideration is that during the time step interval subroutine DRV is called four times to set up the equations shown above.

```
73:C     START EXECUTABLE CODE
74:C
75:C     INTEGRATION COUNTER
76:      I = 1
77:C     HALF OF DELTA TIME
78:      DLTO2 = DLT/2.
79:C
```

```
 80:C     FIRST SET OF DERIVATIVES
 81:      CALL DRV ( I, T, XI, XD(1,1) )
```

Since XI( ) and XD( ) go up to XI(20) and XD(20) in DRVGT, N must be at least 20.

```
 83:C     FIRST INTERMEDIATE INTEGRAL
 84:      DO 1000  K = 1,N
 85:          IF (ABS(XD(K,1)) .LT. SMALL) XD(K,1) = 0.
 86:          XIP(K) = XI(K) + XD(K,1)*DLTO2
 87: 1000     CONTINUE
 88:C     ENDDO
 89:C
 90:C     HALF TIME STEP
 91:      T = T + DLTO2
 92:C
 93:C     INTEGRATION COUNTER
 94:      I = I + 1
 95:C
 96:C     UPDATE TARGET - MISSILE GEOMETRY
 97:      TD = T + TOL
 98:      CALL RELTGT ( TD, DLTO2 )
 99:C
100:C     2ND SET DERIVATIVES
101:      CALL DRV ( I, T, XIP, XD(1,2) )
102:C
103:C     2ND INTERMEDIATE INTEGRALS
104:      DO 2000  K=1,N
105:          IF (ABS(XD(K,2)) .LT. SMALL) XD(K,2) = 0.
106:          XIP(K) = XI(K) + XD(K,2)*DLTO2
107: 2000     CONTINUE
108:C     ENDDO
109:C
110:C     INCREMENT INTEGRATION COUNTER
111:      I = I + 1
112:C
113:C     3RD SET OF DERIVATIVES
114:      CALL DRV ( I, T, XIP, XD(1,3) )
115:C
116:C     3RD INTERMEDIATE VALUES
117:      DO 3000  K=1,N
118:          IF (ABS(XD(K,3)) .LT. SMALL) XD(K,3) = 0.
119:          XIP(K) = XI(K) + XD(K,3)*DLT
120: 3000     CONTINUE
121:C     ENDDO
122:C
123:C     STEP TIME TO END OF INTERVAL
124:      T = T + DLTO2
125:      I = I + 1
126:C
127:C     UPDATE TARGET - MISSILE GEOMETRY
128:      TD = T + TOL
129:      CALL RELTGT ( TD, DLT )
130:C
131:C     4TH SET OF DERIVATIVES
132:      CALL DRV ( I, T, XIP, XD(1,4) )
133:C
134:C     INTEGEATION
135:      DO 4000  K=1,N
136:          IF (ABS(XD(K,4)) .LT. SMALL) XD(K,4) = 0.
137:          XI(K) = XI(K) + DLT*(XD(K,1) + 2.0*(XD(K,2) + XD(K,3))
138:     1                                              + XD(K,4))/6.0
139: 4000     CONTINUE
140:C     ENDDO
141:C
```

```
142:      RETURN
143:      END
```

Subroutine DRVGJ

SEE CLASSIFIED ADDENDUM

Subroutine ANGTRN computes the equivalent gimbal angles in a coordinate system rolled by the angle ANGL.

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM.

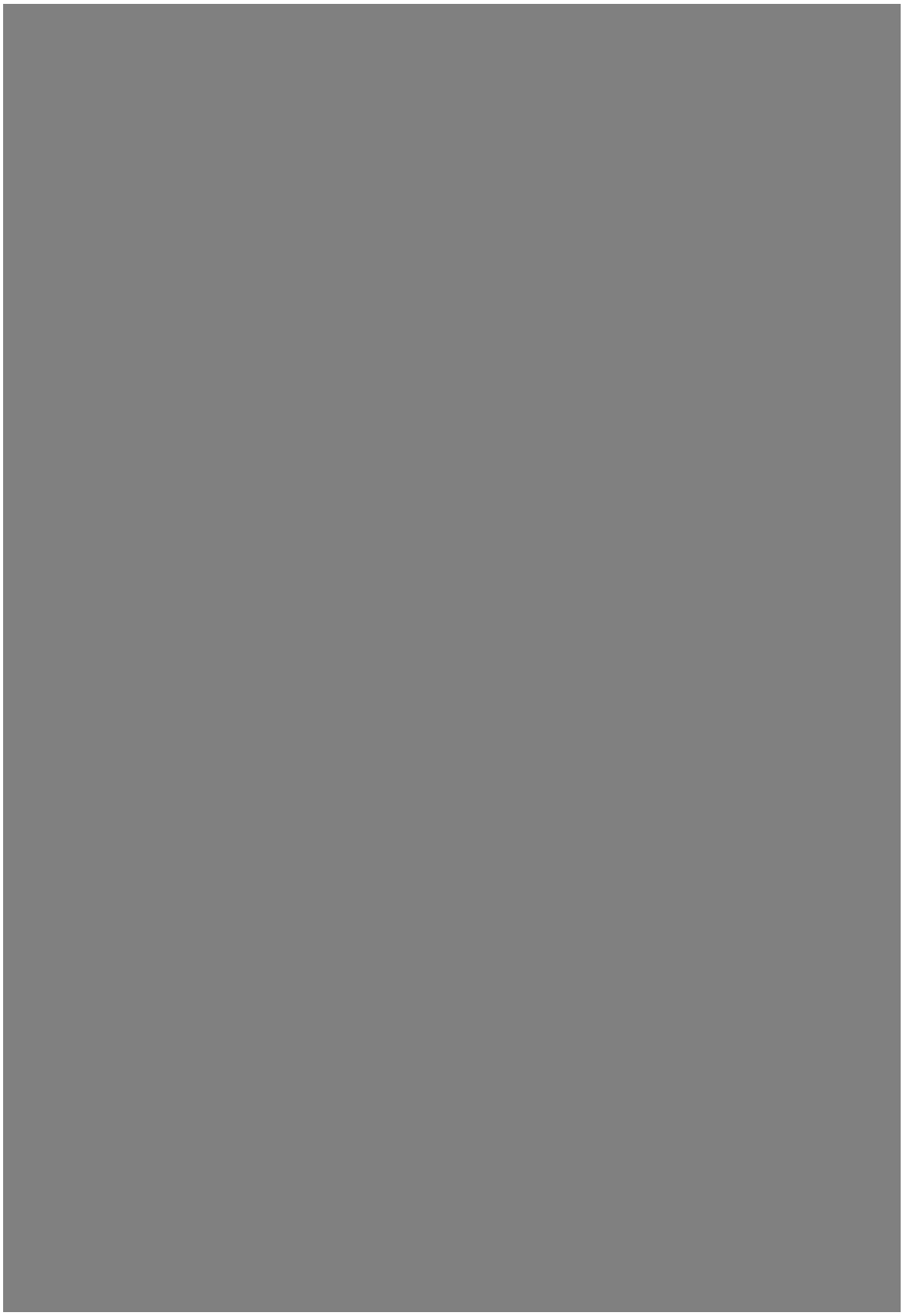
SEE CLASSIFIED ADDENDUM

The terms ANGERP, ANGERY, THEN and THAN are no longer used, since noise and ECM corruption are captured in ELERR and AZERR through employment of the GRAM code.

Phi-rolled puts the seeker frame at 45° angles to the standard azimuth and elevation channels.

SEE CLASSIFIED ADDENDUM

Either a wide or narrow bandwidth can be selected.



SEE CLASSIFIED ADDENDUM

The output of filter 2 in figure 2.32.6.2-1 is

SEE CLASSIFIED ADDENDUM

These outputs use the form

$$y=y_1 + K_0u = y_1 + .0264u \qquad [2.32\text{-}30]$$

This form is developed above line 533 in the annotated code.

SEE CLASSIFIED ADDENDUM

The development for block 3 in figure 2.32.6.2-1 proceeds as follows:

SEE CLASSIFIED ADDENDUM

The total output of Block 3 is

SEE CLASSIFIED ADDENDUM

Filter block 4 of figure 2.32.6.2-1 is developed as follows:

SEE CLASSIFIED ADDENDUM

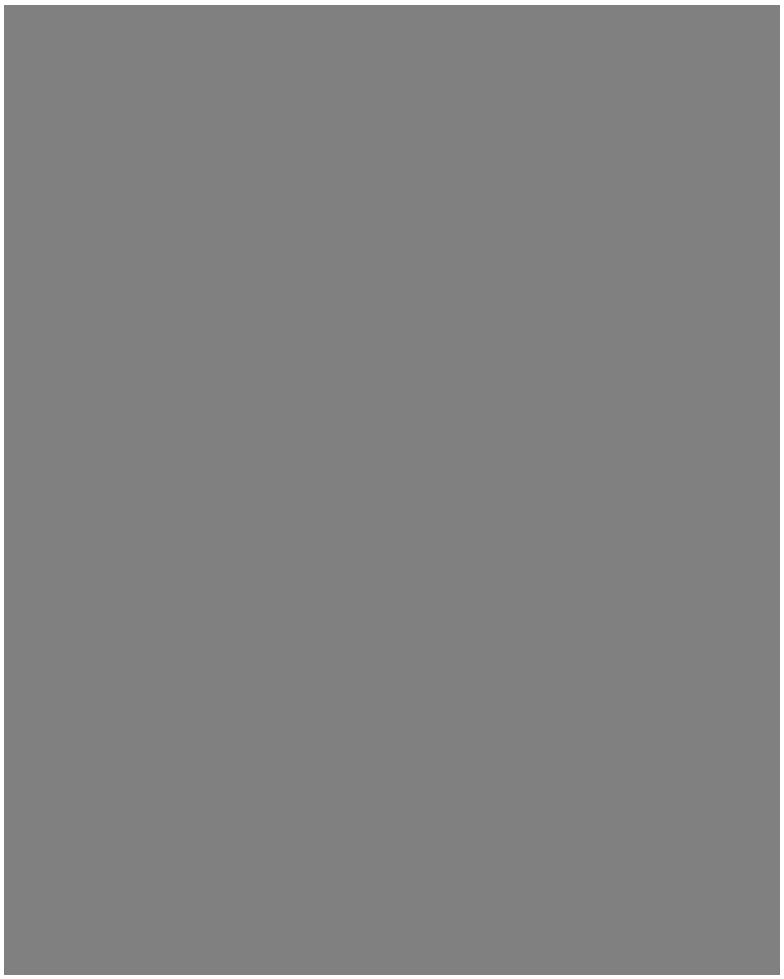
The limited output - XI (9) and XI (10) are limited in the code—equates to

```
      YE11 = XI(9) + GC(17) * YE10
      YA11 = XI(10) + GC(17) * YA10
C
C FIND THE SEEKER ANGLE ACCELERATION WHICH IS MADE UP OF
C THE LIMITED OUTPUT OF BLOCK 4 AND THE FEEDBACK VALUE OF
C THE SEEKER RATES
      YE13 = YE11 - YE21
      YA13 = YA11 - YA21
C
C INITIALIZE THE STATE SPACE EQUATIONS DURING THE FIRST PASS THRU
      IF (KST .EQ. 1 .AND. INTFLG .EQ. 1) THEN
         XI(11) = 0.0
         XI(12) = 0.0
         XI(13) = PGIM
         XI(14) = YGIM
      ENDIF
C
C FIND THE OUTPUT OF BLOCK 5 WHICH IS THE SEEKER ANGLE RATES
      PGIMR = XI(11)
      YGIMR = XI(12)
C
C COMPUTE THE FEEDBACK VALUES BASED ON THE CURRENT VALUES WHICH
C WILL BE USED DURING THE NEXT TIME INCREMENT
```

```
436:      YE21 = GC(18) * PGIMR
437:      YA21 = GC(18) * YGIMR
438:      YE22 = GC(19) * PGIMR
439:      YA22 = GC(19) * YGIMR
440:      YE23 = GC(20) * YE22
441:      YA23 = GC(20) * YA22
442:C
443:C INTEGRATE AND LIMIT THE SEEKER RATES TO DETERMINE THE SEEKER
444:C ANGLES. THESE VALUES ARE USED THROUGHOUT THE GUIDANCE COMPUTER
445:C SIMULATION.
446:      IF (ABS(XI(13)) .GT. GC(48)) XI(13) = SIGN (GC(48), XI(13))
447:      IF (ABS(XI(14)) .GT. GC(48)) XI(14) = SIGN (GC(48), XI(14))
448:C
449:      PGIM = XI(13)
450:      YGIM = XI(14)
451:C
452:      ANGL = PHI
453:      CALL ANGTRN(PGIM,YGIM,PGIMAN,YGIMAN,4,ANGL)
454:C
455:C SET GAIN KT FOR TACHOMETER VELOCITY SCALING POT
456:C AND GAIN KG FOR GYRO VELOCITY SCALING POT
457:      IF (IHMODE .EQ. 1) THEN
458:          VCTMP = GC(63)
459:      ELSE
460:          VCTMP = VCLOSE
461:      ENDIF
462:      VC = ABS(VCTMP) - GC(47)
463:      IF (VC .LE. 0.0) THEN
464:          KT = GC(21) - (VC * GC(23))
465:          KG = GC(22) + (VC * GC(24))
466:      ELSE
467:          KT = GC(21) - (VC * GC(25))
468:          KG = GC(22) + (VC * GC(26))
469:      ENDIF
470:C
471:C IF T IS LESS THAN TGUIDE (GC(2) SECONDS) THEN THE ROTATED GIMBAL
472:C OUTPUT IS NOT USED AND THE INPUT TO THE KG GAIN BLOCK IS
473:C GROUNDED
474:      IF (T .LT. GC(2)) THEN
475:          YE28 = KT*YE22
476:          YA28 = KT*YA22
477:      ELSE
478:          YE28 = VGYROE*KG + KT*YE22
479:          YA28 = VGYROA*KG + KT*YA22
480:      ENDIF
481:C
482:C ONCE THE VARIABLE GAIN ,K1, IS FOUND THE INPUT TO THE WIDE AND
483:C NARROWBAND NAVIGATION COMPUTER CAN BE FOUND
484:      YE30 = YE28 * K1
485:      YA30 = YA28 * K1
486:C
487:C INITIALIZE THE STATE SPACE EQUATIONS DURING THE FIRST PASS THRU
488:      IF (KST .EQ. 1 .AND. INTFLG .EQ. 1) THEN
489:          XI(15) = YE30
490:          XI(16) = YA30
491:          XI(17) = YE30
492:          XI(18) = YA30
493:      ENDIF
494:C
495:C THE OUTPUT OF THE NAVIGATION COMPUTER IS FOUND BASED ON THE
496:C DYMANIC PRESSURE AND THE TIME TO ARM
497:      IF ((TTARM .LT. GC(3) .OR. T .GT. TTARM) .AND.
498:     1       DYNPC .GT. GC(30)) THEN
499:C
```

```
500:C         USE WIDE BANDWIDTH
501:          YE33  = XI(15)
502:          YA33  = XI(16)
503:      ELSE
504:C
505:C         USE NARROW BANDWIDTH
506:          YE33  = XI(17)
507:          YA33  = XI(18)
508:      ENDIF
509:C
510:C INITIALIZE THE STATE SPACE EQUATIONS DURING THE FIRST PASS THRU
511:      IF (KST .EQ. 1 .AND. INTFLG .EQ. 1) THEN
512:          XI(19) = GC(31) * YE33
513:          XI(20) = GC(31) * YA33
514:      ENDIF
515:C
516:C DETERMINE THE GUIDANCE COMMANDS IN THE PITCH AND YAW PLANES BASED
517:C ON THE OUTPUT OF THE GUIDANCE COMPUTER
518:      YE34   = XI(19)
519:      YA34   = XI(20)
520:C
521:C CONVERT THE GUIDANCE COMMANDS WHICH ARE IN VOLTS INTO
522:C G'S
523:      GDCMP = YE34 * GC(32)
524:      GDCMY = YA34 * GC(32)
525:      IF (T .LT. GC(2)) THEN
526:          GDCMP = 0.0
527:          GDCMY = 0.0
528:      ENDIF
529:C
530:C ELEVATION AND AZIMUTH FILTER STATE SPACE EQUATIONS
531:      XD(1)  = -GC(33) * XI(1) + GC(33) * YE04
532:      XD(2)  = -GC(33) * XI(2) + GC(33) * YA04
```

The development of filter 2 of figure 2.32.6.2-1 proceeds as follows:



SEE CLASSIFIED ADDENDUM

Since there is a channel for phi-rolled azimuth and a channel for phi-rolled elevation, there are two sets of coupled differential equations in the code as follows:

```
533:      XD(3)  = XI(4) + GC(10) * YE06
534:      XD(4)  = -GC(34) * XI(3) - GC(35) * XI(4) + GC(36) * YE06
535:      XD(5)  = XI(6) + GC(10) * YA06
536:      XD(6)  = -GC(34) * XI(5) - GC(35) * XI(6) + GC(36) * YA06
537:      XD(7)  = -GC(37) * XI(7) + GC(38) * YE07
538:      XD(8)  = -GC(37) * XI(8) + GC(38) * YA07
539:      XD(9)  = -GC(39) * XI(9) + GC(40) * YE10
540:      XD(10) = -GC(39) * XI(10) + GC(40) * YA10
```

SEE CLASSIFIED ADDENDUM

```
541:      XD(11) = GC(41) * YE13
542:      XD(12) = GC(41) * YA13
543:      XD(13) = PGIMR
544:      XD(14) = YGIMR
```

SEE CLASSIFIED ADDENDUM

```
545:      XD(15) = -GC(42) * XI(15) + GC(42) * YE30
546:      XD(16) = -GC(42) * XI(16) + GC(42) * YA30
```

SEE CLASSIFIED ADDENDUM

```
547:      XD(17) = -GC(43) * XI(17) + GC(43) * YE30
548:      XD(18) = -GC(43) * XI(18) + GC(43) * YA30
```

SEE CLASSIFIED ADDENDUM

```
549:      XD(19) = -GC(44) * XI(19) + GC(45) * YE33
550:      XD(20) = -GC(44) * XI(20) + GC(45) * YA33
```

The differential equations for the three transfer functions above are developed as follows:

SEE CLASSIFIED ADDENDUM

$$\tau \dot{y}out + yout = a\ yin \qquad [2.32\text{-}31]$$

$$\dot{y}out = \frac{1}{\tau}(a\ yin - yout) \qquad [2.32\text{-}32]$$

$$\dot{y}out = \tau_1(a\ yin - yout) \qquad [2.32\text{-}33]$$

```
551:C    *disable guidance is so required
 552:      IF (NOGUID .GT. 0) THEN
 553:         GDCMP=0.0
 554:         GDCMY=0.0
 555:      ENDIF
 556:C
 557:      RETURN
 558:      END
```

2.32.7 Known Problems and Anomolies

There are no known problems at this time.

2.32.8 Software Drivers

An off-line driver was constructed for use in ESAMS 2.6.SMART. It is designed to exercise the guidance algorithms, and it should be applicable to ESAMS 2.7.